

Automatic Fruit and Vegetable Classification Using Deep Learning and Computer Vision: An Approach for Intelligent Weighing Systems

Daniel Brandão Lloyd, Thiago Novaes Borsoni, Bernardo de Oliveira Pinto, Rigel P. Fernandes[✉], Talita V. Ribeiro[✉] and Clayton J. A. Silva

Abstract—We train a YOLOv8s classifier on the *Fruits and Vegetables Detection for YOLOv4* dataset for use in intelligent weighing systems, under an eight-class taxonomy and a fourteen-class one separating bagged from unbagged produce, and obtain near-perfect validation scores. Rather than reporting this as a result, we test whether it is an artifact. Because the source dataset was built for detection and contains bursts of near-identical frames of the same item, an image-level split can place almost identical frames on both sides. We detect near-duplicates by perceptual hashing and re-evaluate under a group-aware split confining every frame of an item to one side. Accuracy moves from 0.999 to 0.995 (eight classes) and stays at 1.000 (fourteen classes), so leakage does not explain the score. Residual errors are semantically coherent — chilli with lemon and apple with tomato — indicating a model that learned appearance. We conclude the benchmark is saturated and that the open problem is domain shift, not classifier capacity. We report single-image inference latency of at most 216ms on a laptop CPU for a 9.81 MB checkpoint, with accuracy stable under illumination and compression changes but degrading under defocus.

Keywords—Computer Vision, Deep Learning, Classification, YOLOv8s, Benchmark Saturation.

I. INTRODUCTION

Automatic classification of fruits and vegetables plays a key role in intelligent weighing systems and retail automation. Visual identification removes the need for manual product selection, reducing errors and waiting times, but real deployments must cope with lighting variation, reflections from plastic bags, overlapping items, and inconsistent camera perspectives, which call for lightweight models balancing precision and efficiency [1]. Compact CNNs such as MobileNetV2 and EfficientNet perform well when fine-tuned on domain-specific datasets [2], colour information improves robustness under translucency and occlusion [3], and YOLO architectures enable real-time inference on embedded hardware, though dataset imbalance and limited cross-domain generalization remain open [4], [5], [6], [7], [8].

Daniel Brandão Lloyd, Thiago Novaes Borsoni, Bernardo de Oliveira Pinto, Rigel P. Fernandes and Clayton J. A. Silva are affiliated with the Data Science and Artificial Intelligence course at IBMEC-RJ (Centro Universitário IBMEC, Rio de Janeiro, Brazil). Emails: {danielllloydaus, thiago.borsoni, bernardopinto984, rigelfernandes}@gmail.com, clayton.silva@ibmec.edu.br.

Talita V. Ribeiro is affiliated with the Data Science and Artificial Intelligence course at IBMEC-RJ and with the Computer Science Course at UERJ. Email: talita.ribeiro@ime.uerj.br.

Our starting point was an ordinary application: fine-tuning a compact classifier for a weighing kiosk. The classifier reached near-perfect validation accuracy after a single epoch, which is not a credible outcome for a recognition task and demanded explanation rather than celebration. This paper is that investigation. We test the most plausible explanation, leakage through near-duplicate frames, with a group-aware evaluation protocol and show it does not account for the result; we then characterize what the model is sensitive to, through error analysis and controlled perturbations, and report latency on CPU hardware comparable to a kiosk. The conclusion is that the benchmark is saturated, which is more useful to the community than another near-perfect score.

A. Related Work

Produce recognition for retail has largely converged on compact CNNs and transfer learning. Rojas-Aranda et al. [2] classify bagged and unbagged fruit with a lightweight CNN augmented by colour features, and feature-fusion approaches [3] report similar gains under translucency. Several works push reported accuracy above 0.99 on curated datasets [4], [7], and reviews of the area [5] catalogue this trend across datasets and architectures. A recurring concern in that literature is dataset quality itself: annotation technique [6] and more challenging, less constrained datasets [8] are proposed precisely because curated benchmarks stop discriminating between models. Our contribution is orthogonal to architectural refinement and sits in that second line: rather than proposing another model, we show that on a repurposed detection dataset near-perfect accuracy survives even a leakage-controlled evaluation, and argue the limiting factor is domain shift rather than classifier capacity, a factor most of the above do not isolate because they evaluate within a single acquisition domain.

II. PROBLEM DEFINITION

The task is multi-class image classification: each frame from the weighing camera must be assigned to one product category, under three constraints. **Latency** must preserve the interaction; we measure on a laptop CPU rather than a GPU, as kiosks rarely ship with discrete accelerators. **Memory** must suit embedded platforms; our checkpoint occupies 9.81 MB. **Visual variability** covers bag translucency, occlusion and lighting, which we quantify with controlled perturbations in Section IV.

III. METHODOLOGY

A. Dataset and Taxonomies

The Fruits and Vegetables Detection for YOLOv4 dataset [1] yields 4 592 usable images after parsing. Product labels are extracted from filenames by a regular-expression parser, and each image receives a binary attribute indicating whether the produce is bagged (wb) or unbagged (wob). The eight-class taxonomy uses product type alone (*apple*, *banana*, *blackberries*, *chilli*, *grapes*, *lemon*, *raspberry*, *tomato*), 140–728 images per class; the fourteen-class taxonomy splits six products into with-bag and without-bag variants, keeping *raspberry* and *blackberries* unbagged only, 140–378 per class. Of 4 592 images, exactly 315 carry no bag marker, and these are precisely the 175 *raspberry* and 140 *blackberries* images: the heuristic leaves no ambiguous case among the six products having both variants, which bounds the label noise it can introduce.

B. Near-Duplicate Grouping

The source dataset was assembled for detection and contains bursts of frames capturing the same physical item, rearranged on the same scale under identical lighting (Fig. 1). An image-level split places almost identical frames on both sides, so a classifier could score perfectly by memorizing items rather than learning appearance.

To test this, we group images before splitting. For each image we compute a 64-bit difference hash on a 9×8 grayscale reduction and, within each class, connect any pair whose Hamming distance is at most eight bits. We additionally connect images sharing a filename prefix once a single trailing counter is removed, but only when that prefix covers at most 30% of the class, so a prefix identifying the class rather than an item is discarded. Connected components give 871 groups (eight classes) and 1 023 (fourteen) from 4 592 images, roughly five frames each; we inspected the largest groups visually to confirm they contain a single item.

Groups are the indivisible unit of a stratified group-wise partition (StratifiedGroupKFold, $k = 3$). Because group sizes are uneven, the realized partition is 2 743/1 849 images (59.7%/40.3%) for eight classes and 2 786/1 806 (60.7%/39.3%) for fourteen; we report measured values rather than a nominal target. An image-level random split (70%/30%, 1 378 validation images) is retained as baseline.

C. Training Setup

A pretrained YOLOv8s classifier (yolov8s-cls.pt) was fine-tuned for 50 epochs, batch size 16, input 224×224 , AdamW, on CPU. All four configurations (two taxonomies $\times$ two splits) share hyperparameters and seed (42), so differences in Table I are attributable to the split alone. Convergence is immediate: validation accuracy exceeded 0.99 after the first epoch in every configuration, which is itself part of the finding, and the remaining epochs leave it essentially unchanged.



Fig. 1. Sample frames from the six largest near-duplicate groups. Each row contains images assigned to a single group: the same physical items rearranged on the same scale. An image-level split would distribute each row across training and validation.

IV. RESULTS

A. Leakage Does Not Explain the Result

The group-aware protocol is substantially stricter: it removes near-duplicate frames across the partition and, as a side effect of uneven group sizes, enlarges the validation set from 30% to roughly 40% of the data. Despite this, Table I shows accuracy falling only from 0.999 to 0.995 for eight classes and not moving at all for fourteen. The leakage hypothesis is therefore not supported, worth stating plainly since it was our own leading explanation and the most natural one given how the dataset was built.

TABLE I. Validation performance under an image-level random split and a group-aware split confining near-duplicate frames of an item to one side.

Taxonomy	Split	Val. img.	Accuracy	Macro F1
8 classes	random	1 378	0.999	0.999
8 classes	grouped	1 849	0.995	0.996
14 classes	random	1 378	1.000	1.000
14 classes	grouped	1 806	1.000	1.000

B. Error Structure

The ten errors in the eight-class group-aware configuration are semantically coherent rather than arbitrary: five *chilli* images predicted as *lemon* and three in reverse, two *apple* predicted as *tomato* (Fig. 2). Both pairs are confusable in shape and colour, chillies and lemons small, elongated, yellow-green; apples and tomatoes round and red. A model that had memorized files would not fail along these axes, so the error structure is evidence the classifier learned appearance. The errors are confined to just two confusable pairs: the remaining

four classes (*banana*, *blackberries*, *grapes*, *raspberry*) receive and emit no misclassification and therefore attain perfect per-class precision, recall and F1, so the sub-unity macro F1 in Table I is driven entirely by the chilli/lemon and apple/tomato confusions rather than spread across the taxonomy. The fourteen-class configuration produces no errors, and its confusion matrix is perfectly diagonal. That the finer taxonomy outperforms the coarser one is counterintuitive: grouping is computed within each class, so the fourteen-class taxonomy yields a different partition, and the two configurations are not evaluated on the same validation images. In a regime where both are near the ceiling, this sampling difference is enough to account for the gap.

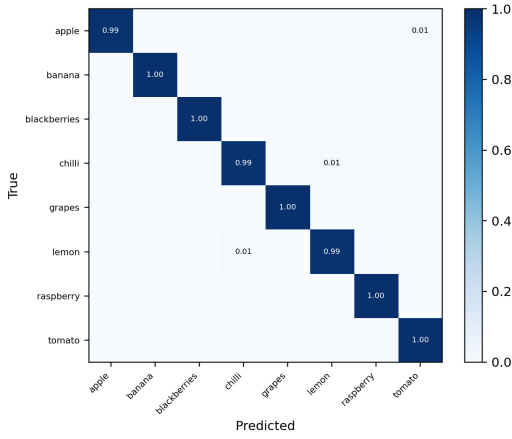


Fig. 2. Row-normalized confusion matrix, eight classes, group-aware split. Off-diagonal mass falls only on the chilli/lemon and apple/tomato pairs.

C. Latency and Sensitivity

Single-image inference on an Intel Core i5-1345U CPU (batch size 1, 224×224 input, 50 timed runs after 10 warm-up runs) averages 204.0 ± 13.0 ms for eight classes and 216.4 ± 21.3 ms for fourteen, with P95 of 223 ms and 261 ms respectively. With a 9.81 MB checkpoint, the model fits the interaction budget of a kiosk without a discrete GPU.

Table II reports accuracy under six controlled perturbations applied to the validation set of each configuration. Brightening, darkening and aggressive JPEG compression leave performance essentially unchanged, while Gaussian blur produces the largest degradation and occlusion a smaller one. For a kiosk this suggests camera focus deserves more attention than illumination control.

TABLE II. Accuracy under six controlled perturbations, group-aware split, on the validation set of each configuration.

Perturbation	8 classes	14 classes
Original	0.993	1.000
Brightened	0.993	1.000
Darkened	0.978	1.000
Occlusion	0.988	0.998
Gaussian blur	0.985	0.980
JPEG (quality 30)	0.995	1.000

V. THREATS TO VALIDITY AND CONCLUSIONS

All images come from a single acquisition campaign: one scale, one camera position, one lighting setup. Group-aware splitting removes leakage between near-duplicate frames but cannot create variability the dataset never had, and nothing reported here speaks to performance under a different camera or store. The grouping procedure is itself heuristic: an eight-bit Hamming threshold may merge similar but distinct items, or fail to merge one item seen from very different angles. Merging too aggressively makes the split stricter, so this bias is conservative with respect to our conclusion; the opposite failure would not be, and we mitigated it by inspecting the largest groups. The bag attribute is filename-derived without manual verification; the pipeline provides a stratified audit routine to quantify its agreement, which we leave as immediate future work.

We set out to explain a near-perfect validation score and found that the most plausible explanation does not hold. A protocol confining all near-duplicate frames of an item to one side of the partition, while enlarging validation to about 40% of the data, leaves accuracy essentially unchanged: 0.995 for eight classes, 1.000 for fourteen, with residual errors following shape and colour similarity as a classifier that learned appearance should produce. We therefore conclude this benchmark is saturated: it no longer distinguishes between models, and near-perfect numbers on it, ours included, carry little information. The useful open problem is elsewhere, since the dataset’s uniform acquisition conditions mean nothing here predicts behaviour under a different camera or store. Future work should prioritize cross-dataset evaluation and multi-source collection over architectural refinement, alongside embedded benchmarking and multi-item scenes. The pipeline including the grouping procedure is available, so the protocol can be applied to other detection datasets repurposed for classification, where the same risk arises even if, as here, it is not the binding constraint.

REFERENCES

- [1] K. Patel, “Fruits & vegetable detection for YOLOv4,” Kaggle dataset, 2021, accessed: 2026-08-07. [Online]. Available: <https://www.kaggle.com/datasets/kvnpatel/fruits-vegetable-detection-for-yolov4>
- [2] J. L. Rojas-Aranda, J. I. Nunez-Varela, J. C. Cuevas-Tello, and G. Rangel-Ramirez, “Fruit classification for retail stores using deep learning,” in *Pattern Recognition (MCPR 2020)*, ser. Lecture Notes in Computer Science, vol. 12088. Springer, Cham, 2020, pp. 3–13.
- [3] H. S. Gill, G. Murugesan, A. Mehbodniya, G. S. Sajja, G. Gupta, and A. Bhatt, “Fruit type classification using deep learning and feature fusion,” *Computers and Electronics in Agriculture*, vol. 211, p. 107990, 2023.
- [4] M. S. Hossain, M. Al-Hammadi, and G. Muhammad, “Automatic fruit classification using deep learning for industrial applications,” *IEEE Transactions on Industrial Informatics*, vol. 15, no. 2, pp. 1027–1034, 2019.
- [5] Y. Gulzar, Z. Ünal, S. Ayoub, F. A. Reegu, and A. Altulih, “Adaptability of deep learning: datasets and strategies in fruit classification,” in *BIO Web of Conferences*, vol. 85. EDP Sciences, 2024, p. 01020.
- [6] N. Mamat, M. F. Othman, R. Abdulghafar, A. A. Alwan, and Y. Gulzar, “Enhancing image annotation technique of fruit classification using a deep learning approach,” *Sustainability*, vol. 15, no. 2, p. 901, 2023.
- [7] J. L. Joseph, V. A. Kumar, and S. P. Mathew, “Fruit classification using deep learning,” in *Innovations in Electrical and Electronic Engineering: Proceedings of ICEEE 2021*. Springer, 2021, pp. 807–817.
- [8] S. Khanna, C. Chattopadhyay, and S. Kundu, “Enhancing fruit and vegetable detection in unconstrained environment with a novel dataset,” *Scientia Horticulturae*, vol. 338, p. 113580, Dec. 2024.